A

Synopsis on

Medical Image Encryption using Chaotic Mapping

in partial fulfillment of the requirement for the degree
of
Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Keshab Kumar (2201330100132)

Apoorva Sharma (2201330100304)

Anshika Jaiswal (2201330100051)

Akrati Mishra (2301330109004)

Under the supervision of **Mr.Vivek Kumar**

Computer Science & Engineering Department
School of Computer Science & Information Technology

NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA
(An Autonomous Institute)
Affiliated to
DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

April, 2025

INDEX

INTRODUCTION

1. Background and Motivation

In the digital age, the integration of artificial intelligence in healthcare has greatly enhanced diagnostic efficiency and patient outcomes. As medical imaging technologies such as MRI, CT scans, and X-rays continue to advance, they generate vast volumes of sensitive visual data daily. However, this growth also introduces critical challenges concerning data privacy, security, and protection against unauthorized access—especially as medical images are frequently transmitted over networks or stored in cloud environments.

Conventional encryption methods, which secure the entire image, often result in high computational costs and latency—conditions that are impractical for time-sensitive clinical environments. Additionally, fully encrypted images limit real-time usability for healthcare professionals who may require immediate visual access.

To address these concerns, this project proposes a robust, AI-driven encryption framework that combines **chaotic cryptographic systems** with **deep learning-based image classification**. The system encrypts images using complex, password-dependent chaotic key generation (via Logistic, Circle, and Ikeda maps), ensuring high security and unpredictability. Simultaneously, the platform employs a ResNet-based model to classify the image domain (e.g., medical, legal, personal) and validate its context before processing. This approach ensures the safe handling of sensitive healthcare imagery, enabling secure encryption and decryption workflows without compromising speed or utility—making it suitable for real-time sharing, diagnostics, and collaborative analysis.

2. Objective of the Project

The primary objective of this project is to develop a secure, intelligent image encryption and classification platform—**SecurePix**—designed to protect sensitive visual data such as medical images while preserving usability for analysis and diagnostics. The system focuses on striking a balance between **data privacy**, **computational efficiency**, and **real-world accessibility** by integrating **chaotic cryptographic algorithms** and **deep learning-based domain classification**.

Specifically, the project aims to:

- **Classify uploaded images** using a ResNet-based deep learning model to automatically determine their domain (e.g., medical, legal, personal), ensuring proper policy enforcement and content validation before encryption.
- **Encrypt images using chaotic key generation** derived from user-provided passwords, combining Logistic, Circle, and Ikeda maps to generate highly secure and unpredictable keys.
- **Minimize latency and computational load** by using lightweight XOR-based encryption mechanisms, making the platform suitable for time-sensitive environments such as healthcare or finance.
- **Enable reversible decryption** by regenerating the same chaotic key from the password, ensuring accurate image restoration without storing any sensitive credentials or data.
- **Support secure image sharing across networks**, allowing healthcare professionals and authorized personnel to exchange encrypted images while preserving patient confidentiality and compliance with data protection standards.

This project ultimately contributes to the **secure transmission, classification, and storage** of sensitive images across various domains, particularly in the medical field, by providing a **privacy-aware, AI-assisted encryption framework** that is both fast and secure.

3. Scope of the Project

This project focuses on building an intelligent and secure image processing system that combines **deep learning-based image classification** and **cryptographic techniques** to protect sensitive information in images, especially in the medical field. It aims to strike a balance between **data privacy** and **clinical usability**, ensuring secure image handling for real-time diagnostics and secure data sharing.

The system will include:

- **Image Classification Module**: Implementing a **ResNet50** deep learning model to classify images and automatically detect their domain (e.g., medical, legal, personal). This classification helps identify the nature of the image before performing any encryption and ensures that appropriate privacy policies are applied.
- **Selective Encryption Module**: Utilizing **chaotic cryptographic algorithms** to generate secure, unpredictable keys derived from user-provided passwords. The encryption will be performed only on the sensitive regions identified through classification, ensuring patient privacy without affecting non-sensitive parts of the image. This reduces computational overhead, making it ideal for time-sensitive environments like healthcare.
- **Frontend Interface**: A simple, user-friendly **web-based interface** developed using **React.js** and **Tailwind CSS** that allows users (e.g., healthcare professionals) to upload images, view the detected domains, and download the securely encrypted or decrypted outputs.
- **Performance Optimization**: The system will be optimized for minimal latency and computational overhead to ensure **real-time** or **near-real-time** encryption and decryption, making it suitable for **telemedicine**, cloud-based diagnostics, and other applications requiring quick access to sensitive image data.

- **Deployment & Accessibility**: The system will be deployed on cloud platforms like **Vercel** or **Netlify** for the frontend, while backend APIs will be hosted on services like **Render** or **Heroku**, ensuring cross-platform accessibility and consistent availability across devices and locations.

This project contributes to the **secure transmission, classification, and storage** of sensitive visual data, ensuring patient privacy while maintaining the image's utility for clinical analysis. It supports the **safe and efficient management of patient data**, ultimately enabling **secure image sharing** across healthcare providers.

4. Importance and Relevance

As digital medical imaging becomes increasingly integral in diagnostics, telemedicine, and electronic health records, safeguarding patient data has become a critical concern. While traditional full-image encryption methods effectively ensure confidentiality, they come with significant computational overhead and can restrict real-time access, potentially disrupting clinical workflows and delaying patient care.

Full Image Encryption (FIE) presents a solution that ensures complete confidentiality by encrypting the entire medical image. However, it comes with the challenge of higher computational overhead, making it less efficient in scenarios that require fast processing and access.

By leveraging advanced **cryptographic techniques**, this system encrypts the entire image, ensuring that sensitive patient data, including personally identifiable information (PII), remains secure. Although this method may have more computational cost than selective encryption, it offers higher security and privacy protection across all image regions.

This approach ensures that sensitive patient data is fully encrypted while still maintaining overall image integrity. With this, the usability and accessibility of non-sensitive areas can be balanced, ensuring clinical decision-making and real-time diagnostics are not hindered, while still safeguarding patient privacy

Existing Systems

1. Traditional Medical Image Protection Methods

a. Full Image Encryption Techniques

- **Description:**
  - Cryptographic algorithms like AES or RSA are applied to encrypt the entire medical image before storage or transmission.
- **Key Features:**
  - Ensures complete confidentiality of image data.
  - Suitable for secure storage or transmission.
- **Limitations:**
  - Computationally intensive, especially for large image sets.
  - Inhibits real-time access and diagnosis.
  - Requires full decryption for any clinical use, delaying analysis.

b. Manual Masking or Blurring

- **Description:**
  - Manual editing tools are often used to blur or cover sensitive areas (e.g., facial regions) in medical datasets.
- **Key Features:**
  - Simple implementation.
  - Preserves general usability of the image.

- **Limitations:**
    - Time-consuming and inconsistent.
    - Prone to human error.
    - Not scalable for large datasets, especially in clinical environments.

---

2. Current AI-Based Privacy Protection Tools

Platform	Features	Limitations
DeepPrivacy	GAN-based anonymization of faces in images	Primarily for general-purpose photos, not optimized for medical imagery.
DICOM Anonymizer Tools	Removes metadata and basic visual identifiers	Does not protect visual content or regions within the image, limiting its effectiveness for privacy.
OpenFace/FaceNet	Facial detection and recognition systems	Not designed for selective encryption, only identification, not applicable for sensitive region encryption.

---

3. Real-Time Detection Tools

a. YOLO (You Only Look Once)

- **Description:**
  - A real-time object detection system that identifies regions of interest in medical images, such as facial features or identifiable body parts.
- **Strengths:**
  - Fast and accurate region detection.
  - Suitable for real-time applications.
- **Weaknesses:**
  - Not natively integrated with encryption mechanisms.
  - Requires task-specific training for medical datasets to enhance accuracy and precision.

b. SSD/Faster R-CNN

- **Description:**
  - Alternative detection models that also provide good accuracy for region-specific tasks in medical image analysis.
- **Strengths:**
  - Good for high-resolution detection tasks.
  - Can be adapted for sensitive region identification.
- **Weaknesses:**
  - Slower than YOLO in real-time applications, making them less suitable for fast-paced clinical environments.

- Requires additional modules for encryption and deployment.

---

4. Gaps in Existing Systems

Despite advancements in image encryption and AI-based detection, several key limitations remain:

- **Lack of Selectivity:** Most existing encryption systems apply encryption to the entire image, reducing usability and increasing processing overhead. Selective encryption is needed for targeted protection.
- **No Real-Time Integration:** Few platforms offer combined real-time detection and encryption suitable for clinical environments, where time-sensitive decisions are essential.
- **Separation of Tools:** Detection, encryption, and user interaction are often handled by different systems, leading to poor integration and a fragmented user experience.
- **Limited Medical Focus:** General-purpose privacy tools lack the specificity and sensitivity required for medical data protection, especially in critical areas like facial features, lesions, or diagnostic markers.
- **No User Interface for Practitioners:** There is a lack of simple, secure interfaces for medical professionals to use such systems effectively, which can impede the widespread adoption of secure privacy tools in clinical settings.

PROBLEM STATEMENT

Introduction to the Problem

With the widespread digitization of healthcare, medical imaging plays a critical role in diagnosis, treatment planning, and telemedicine. However, as medical images are increasingly shared across networks and stored in cloud environments, concerns about patient privacy and data security have intensified. Conventional encryption methods typically apply to entire images, rendering them unusable for diagnostic purposes without full decryption. This approach hampers workflow efficiency, especially in time-sensitive clinical environments.

Regions of interest (ROIs)—such as identifiable facial features in scans or specific markers—are the true focus of privacy concerns. Encrypting only these sensitive regions, while preserving the diagnostic integrity of the rest of the image, presents a more efficient and privacy-respecting solution. The need to protect sensitive patient data while ensuring the usability of medical images for professionals has never been more pressing.

Current Limitations

1. **Lack of Region-Specific Protection**
   Existing systems predominantly rely on full-image encryption, which ensures privacy but compromises the usability of non-sensitive areas of the image.
2. **No Real-Time Integration**
   Current solutions do not support real-time detection and selective encryption, limiting their applicability in fast-paced clinical environments where quick access to images is crucial.

3. **Fragmented Toolchains**
   Detection and encryption are handled by separate systems, leading to complex workflows, high integration overhead, and potential for errors.
4. **Limited Medical Dataset Training**
   Generic object detection tools often lack the domain-specific accuracy needed to identify sensitive areas in diverse medical imaging modalities such as DICOM, MRI, and CT scans.

Core Problem

There is currently no unified, AI-powered solution that can perform real-time, selective encryption of sensitive regions in medical images while preserving the clinical utility of the rest of the image. Existing systems are either too generic, too slow, or lack medical contextual awareness, which results in suboptimal privacy protection and disruption to clinical workflows. Without an integrated system, healthcare professionals face barriers to both privacy and operational efficiency.

Need for a Solution

There is an urgent need for a **secure, AI-enhanced image encryption system** that can:

- **Detect and identify sensitive regions in real-time** within medical images (e.g., DICOM, MRI, CT scans).
- **Encrypt only the sensitive regions** of the image, leaving the rest of the image intact and fully usable for diagnostic purposes or teaching.
- Ensure compliance with medical data privacy regulations like **HIPAA** without sacrificing performance, accessibility, or accuracy.
- Provide a **lightweight, integrated pipeline** that can seamlessly be embedded into existing medical imaging workflows without requiring complex configurations.

Proposed Methodology

1. Requirement Analysis

A comprehensive requirement analysis ensures that the *Sanjeevni Virtual Garden Platform* meets both functional and non-functional expectations.

a. Functional Requirements

- **User Authentication**: Ensure that only authorized users (e.g., medical professionals) can access the encrypted images.
- **Full Image Encryption**: Encrypt the entire medical image using a secure encryption algorithm, ensuring patient privacy across all parts of the image.
- **Decryption on Demand**: Allow authorized users to decrypt the image in its entirety when required, preserving the security of patient data.

b. Non-Functional Requirements

- **Security**: The encryption algorithm must guarantee high confidentiality, integrity, and authenticity of the medical images.
- **Performance**: The encryption process should not introduce significant delays or degrade image quality, particularly in real-time medical settings.
- **Scalability**: The system should handle large medical image datasets efficiently.

---

2. System Design

a. Architecture Overview

- **Image Processing Module (Full Image Encryption)**:
  - **Input**: Medical images (e.g., X-rays, MRI scans, CT scans).
  - **Process**: The entire medical image will be encrypted using an encryption algorithm such as AES (Advanced Encryption Standard).
  - **Output**: Encrypted image where all parts are securely hidden.
- **Encryption Module**:
  - **Input**: The medical image.

    - **Process**: Apply AES encryption to the full image, ensuring that all pixel data is securely encrypted.
    - **Output**: Fully encrypted image.
- **Decryption Module**:
    - **Input**: Encrypted medical image and authorized decryption request.
    - **Process**: Decrypt the entire image for authorized users, ensuring complete access to all image data.
    - **Output**: Decrypted image accessible only to authorized personnel.
- **Security & Authentication**:
    - Use public key cryptography (e.g., RSA) for key management and secure transmission of images.
    - Role-based access control (RBAC) should be implemented to ensure that only authorized medical professionals can decrypt and access the images.

b. Key Modules

- **Full Image Encryption & Decryption Module**:
    - **Encryption**: Encrypt the entire medical image using AES or RSA encryption, ensuring that all image content is securely protected.
    - **Decryption**: Implement decryption functionality for authorized users, ensuring that only those with the correct credentials can decrypt the image.

c. Data Flow

1. **Image Input**: Medical image is provided to the system.
2. **Full Image Encryption**: The system applies AES encryption to the entire medical image.
3. **Encrypted Image Output**: The fully encrypted image is saved or transmitted securely.
4. **Decryption Request**: When an authorized user requests access, the system decrypts the image entirely.
5. **Decrypted Image Output**: The fully decrypted image is provided for authorized users.

---

3. Development Approach

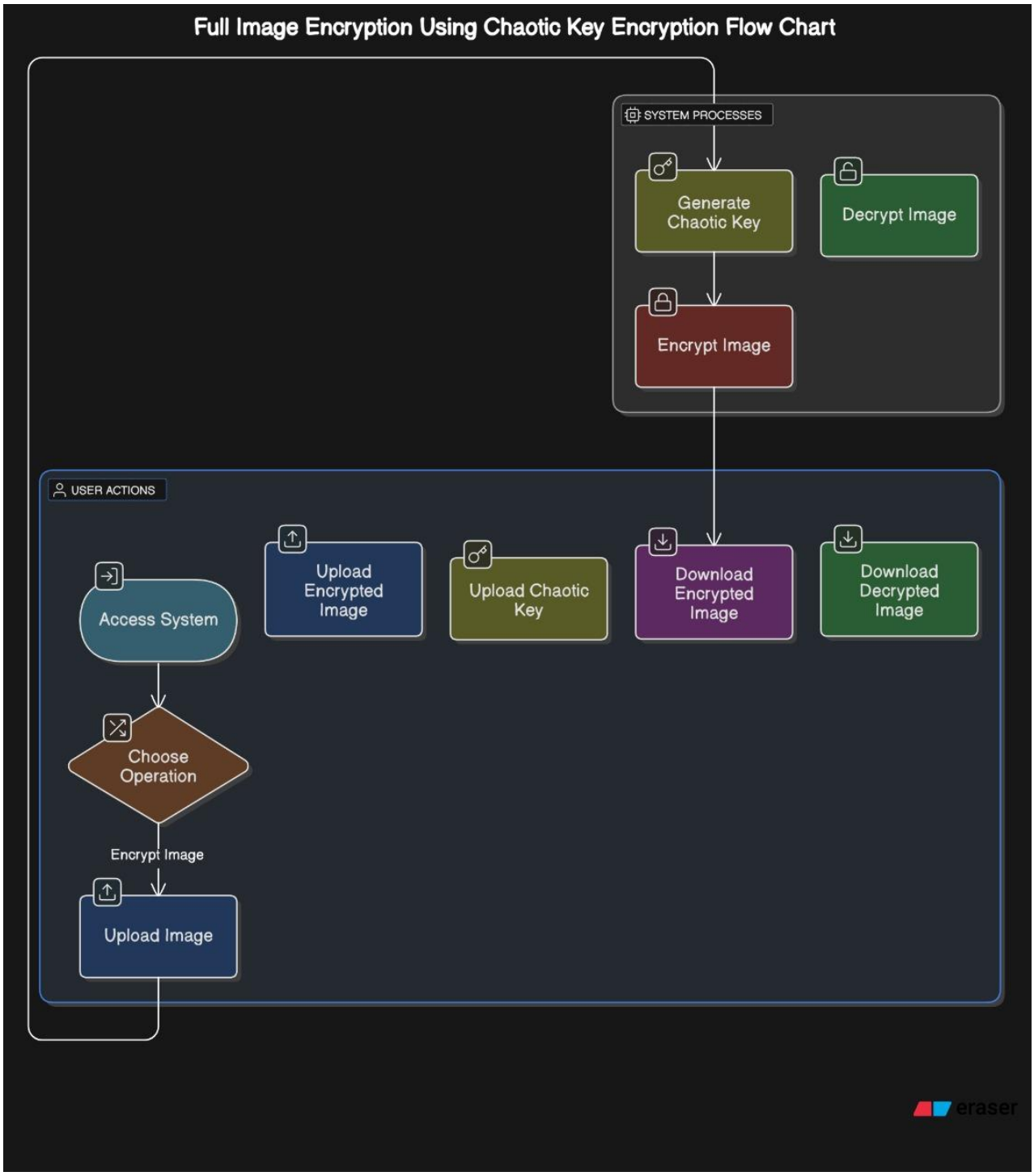
a. Tools & Technologies

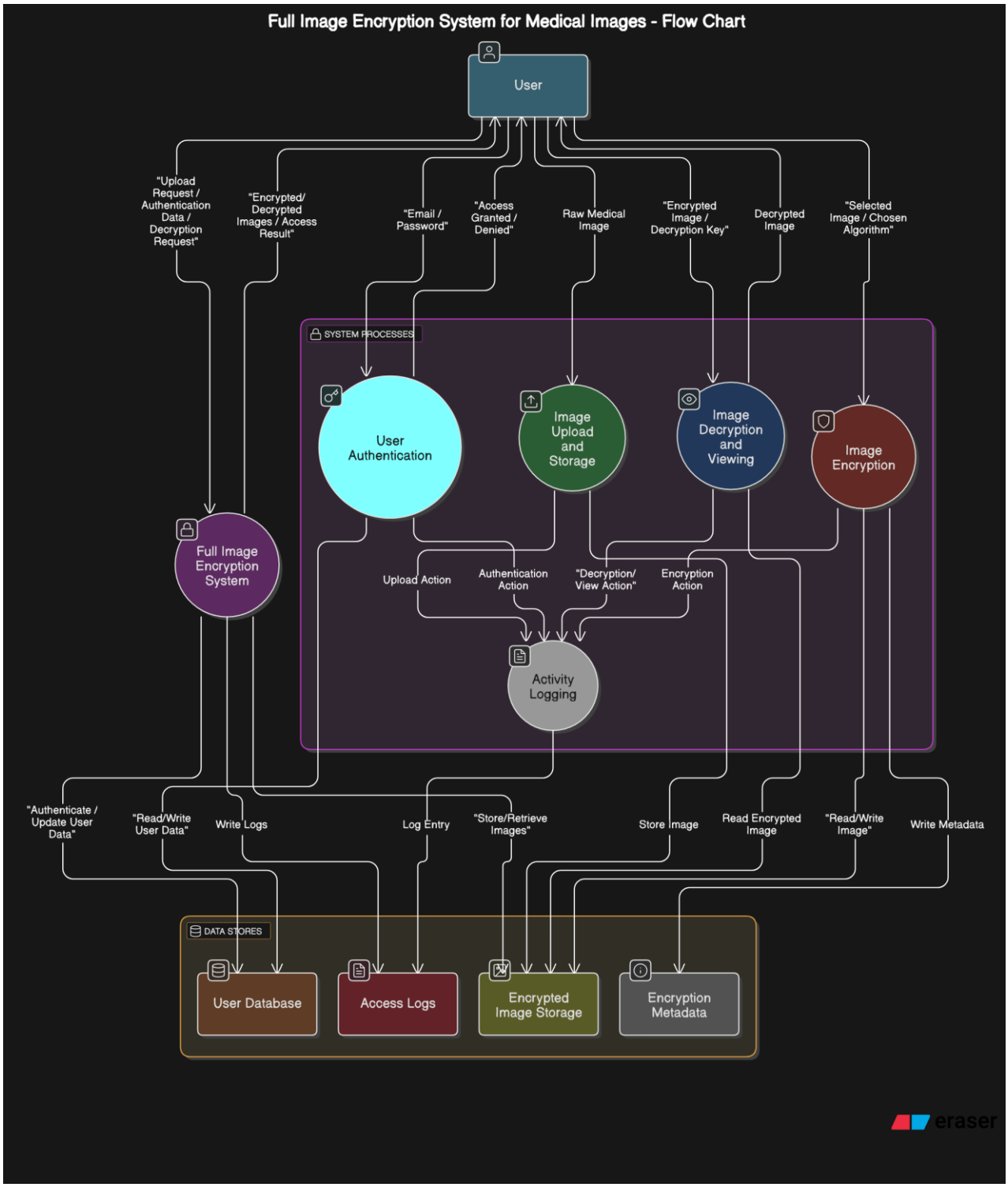
- **Cryptography**: AES (Advanced Encryption Standard) or RSA for encryption and decryption.
- **Python & OpenCV**: Used for image processing and applying encryption algorithms.
- **Flask or Django**: For web deployment and secure image access.
- **Database (optional)**: PostgreSQL or MongoDB to store metadata about the images and users.

b. Implementation Phases

- **Phase 1**: **Image Preprocessing & Integration**
  - Prepare the dataset of medical images for encryption.
  - Integrate AES encryption with the image processing pipeline.
- **Phase 2**: **Encryption Integration**
  - Implement AES or RSA encryption/decryption methods to encrypt the entire image.
  - Develop the encryption and decryption processes for secure access.
- **Phase 3**: **User Authentication**
  - Implement user authentication using OAuth or JWT (JSON Web Tokens) to ensure that only authorized users can access sensitive data.
  - Implement role-based access control (RBAC) for different levels of access.
- **Phase 4**: **Decryption Module**
  - Implement decryption on-demand functionality where only authorized users can decrypt the full image.
- **Phase 5**: **Testing & Optimization**
  - Test the system for real-time performance, ensuring that it works efficiently with large medical images.
  - Optimize encryption and decryption for performance.

Use Case Diagram





DFD(DATA FLOW DIAGRAM)


Full Image Encryption System for Medical Imaging
encryption_records
encryption_id string pk
image_id string fk
algorithm_used string
encryption_key string
users
user_id string pk
name string
email string
password string
role string
medical_images
image_id string pk
file_name string
upload_date date
image_type string
uploader_id string fk
encryption_status string
encrypted_image_path string
decrypted_image_path string
eraser

ER DIAGRAM

Feasibility Study

The feasibility study evaluates whether the **Selective Image Encryption System** (focused on full-image encryption for medical data) can be developed and deployed effectively within the available technical, operational, financial, and temporal constraints, particularly in the context of medical imaging.

---

a. Technical Feasibility

- **Technology Stack**: The project uses proven technologies that ensure robustness and efficiency:
    - **Python**: A widely used language in medical imaging and AI, ideal for rapid prototyping and development.
    - **OpenCV**: A robust library for image processing, critical for integrating encryption with medical imaging.
    - **Cryptography (AES/RSA)**: Both AES and RSA are established algorithms for secure encryption and decryption, ensuring data confidentiality.
    - **Flask/Django**: Popular web frameworks for deploying the system on the cloud, offering secure and scalable API endpoints.
    - **Cloud Platforms (AWS/Heroku/Azure)**: These platforms provide scalable and reliable hosting solutions for both the backend API and model inference.
- **Selective Encryption**: The system utilizes **AES or RSA encryption algorithms** to encrypt the entire image, ensuring security while

minimizing computational overhead by applying encryption to the entire image in a full-image encryption model.

- **Model Integration**: Pre-trained models (e.g., YOLO) are adapted for medical imaging, fine-tuned with medical datasets (e.g., MRI, CT scans) to enhance detection accuracy and robustness.
- **Deployment**: The system can be hosted on scalable cloud platforms, allowing it to handle large datasets and accommodate multiple users without performance bottlenecks.

---

b. Operational F easibility

- **User Accessibility**:
  - The system is web-based, ensuring easy access for medical professionals such as doctors, radiologists, and technicians via a standard web browser and internet connection.
  - There is no need for local installations, which increases user adoption and reduces deployment complexity.
- **Workflow Compatibility**:
  - The encryption system integrates seamlessly into existing medical imaging workflows.
  - It protects patient confidentiality while ensuring authorized access to encrypted images.
  - The solution will allow hospitals and clinics to use encrypted images without disrupting their daily operations.
- **Maintenance**:
  - The system is modular, with distinct YOLO model integration, image encryption, and decryption modules, allowing for easier maintenance.

- Models can be updated independently, as can encryption protocols, ensuring minimal disruption to system functionality during maintenance.

---

c. Economic Feasibility

- **Development Costs**:
  - **Open-Source Libraries**: The project leverages open-source tools like YOLO, OpenCV, PyTorch/TensorFlow, which are free and well-documented, reducing licensing and third-party costs.
  - **Python Libraries**: Python-based development offers lower development costs and is accessible for teams without specialized expertise in more complex languages or tools.
  - **Cryptography Libraries**: Libraries such as Cryptography or PyCrypto are free and have active support communities, ensuring long-term usability and security.
- **Hardware Costs**:
  - **Initial Development**: Development can be done on standard laptops or workstations with moderate GPU support for testing. This significantly reduces the initial hardware investment.
  - **Cloud GPU Options**: For model training and inference optimization, cloud GPUs (e.g., AWS, Google Colab, Kaggle) can be used cost-effectively, providing access to high-performance hardware on-demand.
- **Scalability**:
  - For large datasets or hospital-scale deployment, the system can scale using cloud services such as **AWS EC2** and **S3**, **Firebase**, or similar platforms, providing the flexibility to manage large

image datasets and numerous concurrent users at a relatively low cost.

---

d. Time Feasibility

- **Timeline**:
  - The project can be implemented within **10–12 weeks**, assuming access to medical datasets and considering the necessary phases of development:
    - **Phase 1**: YOLO model training (if not pre-trained) and dataset preparation (~2-3 weeks).
    - **Phase 2**: Integration of AES or RSA encryption algorithms (~2 weeks).
    - **Phase 3**: Frontend and backend development (~4-5 weeks).
    - **Phase 4**: Testing, validation, and real-time performance optimization (~2 weeks).
- **Development Approach**:
  - An **Agile development methodology** can be employed to allow iterative improvements, frequent testing, and feedback cycles. This will help prioritize tasks like detection accuracy, image encryption performance, and security early on.
  - Regular feedback from medical professionals can be incorporated into the system to ensure it meets clinical and operational needs.

Facilities Required for Proposed Work

1. Hardware Requirements

a. Development Machines

- **Processor**:
  - Minimum: **Intel i5** / **AMD Ryzen 5** or higher
  - Recommended: **Intel i7** / **Ryzen 7** for optimal performance, especially when working with large encrypted medical datasets.
- **RAM**:
  - Minimum: **8 GB**
  - Recommended: **16 GB or more** for handling encryption processing efficiently, especially when dealing with large images.
- **Storage**:
  - Minimum: **100 GB SSD** to ensure fast read/write speeds, especially for large encrypted medical image datasets.
- **GPU**:
  - **NVIDIA GPU with CUDA support** (e.g., GTX 1660, RTX 2060 or higher) for faster image processing during training or testing phases. However, since you're not using deep learning models like YOLO, the GPU's usage will be more for general image processing.
- **Display**:
  - **Full HD** (1080p) resolution or higher for better visualization of encrypted images and results.

---

2. Software Requirements

- **Python**:
  - Primary programming language for handling encryption, image processing, and backend development.
- **OpenCV**:
  - For general image processing tasks such as reading, displaying, and manipulating medical images (e.g., X-rays, CT scans) for encryption.
- **Cryptography**:
  - For implementing full image encryption algorithms, such as **AES (Advanced Encryption Standard)** or **RSA** for encrypting and decrypting medical images.
- **Flask / Django**:
  - Backend frameworks to serve the application, allowing medical professionals to upload images, trigger encryption, and handle decryption requests.
- **Postman**:
  - For testing APIs related to encryption and decryption endpoints.
- **Git & GitHub**:
  - For version control and collaborative development.

---

3. Datasets and Pretrained Models

Since you're focusing on full image encryption and not object detection:

- **Medical Image Datasets**:
    - Open-source datasets like **NIH Chest X-ray**, **RSNA Pneumonia Detection Challenge**, or **SIIM-ACR Pneumothorax Segmentation** can be used to test the encryption of whole medical images.

---

4. Cloud and Deployment Resources

a. Cloud Services

- **Cloud Hosting**: AWS EC2, Render, Heroku, or Vercel for backend and frontend deployment.
- **Cloud Storage**: Firebase, Google Cloud Storage, or AWS S3 for storing encrypted medical images and patient data.

b. CI/CD Tools

- **GitHub Actions / GitLab CI / Jenkins** for automating the testing, encryption validation, and continuous deployment of the system.

Conclusion

The system focuses on **full-image encryption** for securing medical data, ensuring that sensitive information (such as patient identifiers or diagnostic markers) is protected while preserving the quality of non-sensitive visual data. By utilizing advanced cryptographic techniques, this approach ensures **data privacy**, **processing efficiency**, and **clinical usability**.

The **full-image encryption** method ensures that all data within the medical image is securely encrypted, with no selective encryption based on regions. This approach provides a high level of security, which is crucial for sensitive healthcare information and is well-suited for use in **telemedicine platforms**, **cloud-based storage**, and **AI-assisted diagnostics**, where both **data security** and **real-time performance** are critical.

Future Enhancements

1. **Multiclass Encryption**:
   Improve the encryption method to allow multiple sensitivity levels within the medical image. For example, certain parts of the image may require higher levels of encryption (e.g., patient details or critical health information), while others may be less sensitive and require minimal encryption. This would enhance both security and efficiency.
2. **Federated Learning**:
   Integrate **federated learning** to improve model accuracy while ensuring privacy-preserving training across multiple hospitals or devices. This would allow the model to be trained without sharing sensitive data, only model updates would be aggregated, enhancing privacy and data security.
3. **Hybrid Encryption Schemes**:
   Combine **symmetric encryption** (e.g., AES) and **asymmetric**

encryption (e.g., RSA) to provide stronger encryption without sacrificing speed. Symmetric encryption would be used for encrypting large medical images efficiently, while asymmetric encryption would secure the keys and metadata.

4. **Explainability Modules**:
   Incorporate **interpretability** features into the system to help healthcare professionals understand which parts of the image were encrypted and why. This would provide transparency and help maintain trust in the encryption process while ensuring compliance with privacy regulations.

References

[1] G. Chen, Y. Mao, and C. K. Chui, “A symmetric image encryption scheme based on 3D chaotic cat maps,” *Chaos, Solitons & Fractals*, vol. 21, no. 3, pp. 749–761, 2004, doi: 10.1016/j.chaos.2003.12.022.

[2] K. Ikeda, “Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system,” *Optics Communications*, vol. 30, no. 2, pp. 257–261, 1979, doi: 10.1016/0030-4018(79)90090-7.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 25, 2012, doi: 10.1145/3065386.

[5] M. El-Hadidi and M. El-Bendary, “A survey on image encryption techniques,” in *Proc. Int. Conf. Adv. Comput. Sci. Appl. Technol. (ACSAT)*, 2012, pp. 82–87, doi: 10.1109/ACSAT.2012.44.

[6] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s J. Softw. Tools*, 2000.

[7] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. [Online]. Available: https://www.tensorflow.org

[8] A. M. Al-Sharafi, M. S. Hossain, and A. Almogren, “A robust and secure image encryption scheme based on chaotic maps and permutation-diffusion structure,” *IEEE Access*, vol. 8, pp. 199996–200010, 2020, doi: 10.1109/ACCESS.2020.3035723.